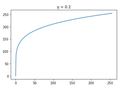
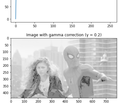
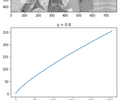
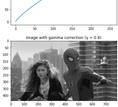
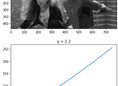
In [ ]:
```
import numpy as np
import cv2 as cv
import matplotlib.pyplot as plt

f = cv.imread('spider.png', cv.IMREAD_GRAYSCALE)
assert f is not None

cv.namedWindow('Image', cv.WINDOW_AUTOSIZE)
cv.imshow('Image', f)
cv.waitKey(0)

img = cv.cvtColor(f, cv.COLOR_BGR2RGB)
fig, ax = plt.subplots()
ax.imshow(img)
ax.set_title('Original Image')
plt.show()


gamma_list = [0.2, 0.8, 1.2, 2]
for gamma in gamma_list:

    t = np.array([(p/255)**gamma*255 for p in range(0,256)]).astype(np.uint8)
    g = cv.LUT(f, t)

    title = "y = " + str(gamma)
    fig, ax = plt.subplots()
    ax.set_title(title)
    ax.plot(t)
    cv.imshow('Image', g)
    cv.waitKey(0)

    fig, ax = plt.subplots()
    img = cv.cvtColor(g, cv.COLOR_BGR2RGB)
    ax.imshow(img)
    title1 = "Image with gamma correction (y = " + str(gamma) + ")"
    ax.set_title(title1)
    plt.show()


cv.destroyAllWindows()
```

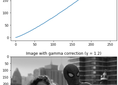


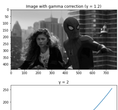


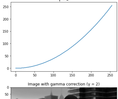




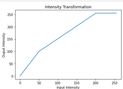


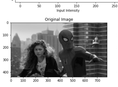




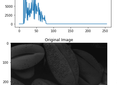


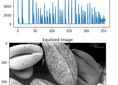
In [ ]:
```
import numpy as np
import cv2 as cv
import matplotlib.pyplot as plt

f = cv.imread('spider.png', cv.IMREAD_GRAYSCALE)
assert f is not None

t1 = np.linspace(0,100,51)
t2 = np.linspace(101, 255, 150)
t3 = np.linspace(255,255,55)

t = np.concatenate((t1, t2, t3), axis=0).astype(np.uint8)

fig, ax = plt.subplots()
ax.plot(t)
ax.set_title('Intensity Transformation')
ax.set_xlabel('Input Intensity')
ax.set_ylabel('Ouput Intensity')
plt.savefig('t.png')
g = cv.LUT(f, t)

cv.namedWindow('Image', cv.WINDOW_AUTOSIZE)
cv.imshow('Image', f)
cv.waitKey(0)
cv.imshow('Image', g)
cv.waitKey(0)
cv.destroyAllWindows()

fig,ax = plt.subplots(2,1,figsize = (12,18))
ax[0].imshow(f, cmap='gray')
plt.imshow(cv.cvtColor(f, cv.COLOR_BGR2RGB))
plt.title('Original Image')
plt.imshow(cv.cvtColor(g, cv.COLOR_BGR2RGB))
plt.title('Intensity Transformation')
plt.show()
```

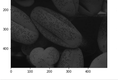


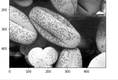


In [ ]:
```
import numpy as np
import cv2 as cv
import matplotlib.pyplot as plt

f = cv.imread('cells.tif',cv.IMREAD_GRAYSCALE)
assert f is not None

hist_f = cv.calcHist([f], [0], None, [256], [0,256])
plt.plot(hist_f)
hist_g = cv.calcHist([g], [0], None, [256], [0,256])

fig,ax = plt.subplots(2,1,figsize = (12,4))
ax[0].plot(hist_f)
plt.title('Histogram of Original Image')
ax[1].plot(hist_g)
plt.title('Histogram of Equilized Image')
plt.show()

fig,ax = plt.subplots(2,1,figsize = (12,8))
ax[0].imshow(cv.cvtColor(f, cv.COLOR_BGR2RGB))
plt.title('Original Image')
ax[1].imshow(cv.cvtColor(g, cv.COLOR_BGR2RGB))
plt.title('Equilized Image')
plt.show()
```



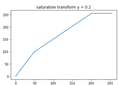






In [ ]:
```
import numpy as np
import cv2 as cv
import matplotlib.pyplot as plt

f = cv.imread('rice_pass.jpg',cv.IMREAD_COLOR)
assert f is not None

hsv = cv.cvtColor(f, cv.COLOR_BGR2HSV)
h, s, v = cv.split(hsv)

t1 = np.linspace(0,100,51)
t2 = np.linspace(101, 255, 150)
t3 = np.linspace(255,255,55)

t = np.concatenate((t1, t2, t3), axis=0).astype(np.uint8)
assert len(t == 256
s_t = cv.LUT(s, t)

fig, ax = plt.subplots()
ax.set_title('Saturation Transformation')
ax.plot(t)


final_hsv = cv.merge((h, s_t, v))
g = cv.cvtColor(final_hsv, cv.COLOR_HSV2BGR)

cv.namedWindow('Image', cv.WINDOW_AUTOSIZE)
cv.imshow('Image', f)
cv.waitKey(0)
cv.imshow('Image', g)
cv.waitKey(0)
cv.destroyAllWindows()

fig,ax = plt.subplots(2,1,figsize = (12,18))
ax[0].imshow(f, cmap='gray')
plt.imshow(cv.cvtColor(f, cv.COLOR_BGR2RGB))
plt.title('Original Image')
ax[1].imshow(g, cmap='gray')
plt.imshow(cv.cvtColor(g, cv.COLOR_BGR2RGB))
plt.title('Saturation Enhanced Image')
```

Out[ ]:
```
Text(0.5, 1.0, 'Saturation Enhanced Image')
```



In [ ]:
```
import numpy as np
import cv2 as cv
import matplotlib.pyplot as plt

f = cv.imread('rice_pass.jpg',cv.IMREAD_COLOR)
assert f is not None

hsv = cv.cvtColor(f, cv.COLOR_BGR2HSV)
h, s, v = cv.split(hsv)

t = np.array([(p/255)**gamma*255 for p in range(0,256)]).astype(np.uint8)
h_t = cv.LUT(h, t)

fig, ax = plt.subplots()
ax.set_title('Hue Transformation')
ax.plot(t)


final_hsv = cv.merge((h_, s, v))
g = cv.cvtColor(final_hsv, cv.COLOR_HSV2BGR)

cv.namedWindow('Image', cv.WINDOW_AUTOSIZE)
cv.imshow('Image', f)
cv.waitKey(0)
cv.imshow('Image', g)
cv.waitKey(0)
cv.destroyAllWindows()

fig,ax = plt.subplots(2,1,figsize = (12,18))
ax[0].imshow(f, cmap='gray')
plt.imshow(cv.cvtColor(f, cv.COLOR_BGR2RGB))
plt.title('Original Image')
ax[1].imshow(g, cmap='gray')
plt.imshow(cv.cvtColor(g, cv.COLOR_BGR2RGB))
plt.title('Hue changed Image')
```

Out[ ]:
```
Text(0.5, 1.0, 'Hue changed Image')
```